regression

#### Yuki Hirayama

#### 2023-05-19

# Data Wrangling  
  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.4.1 ✔ purrr 0.3.4  
## ✔ tibble 3.2.1 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.1

## Warning: package 'tibble' was built under R version 4.2.3

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(tseries)

## Warning: package 'tseries' was built under R version 4.2.3

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(leaps)

## Warning: package 'leaps' was built under R version 4.2.3

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(forecast)

## Warning: package 'forecast' was built under R version 4.2.3

path <- "C:\\Users\\y1u0u\\Documents\\R\\2223\_3\_spring\\01\_TS\\fp"  
df <- read\_csv(file.path(path, 'finalData.csv'))

## Rows: 4018 Columns: 34

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (6): name, preciptype, conditions, description, icon, stations  
## dbl (25): tempmax, tempmin, temp, feelslikemax, feelslikemin, feelslike, de...  
## dttm (2): sunrise, sunset  
## date (1): datetime  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# drop unnecessary variables  
df\_base <- df %>% select(-feelslikemax,  
 -feelslikemin,  
 -feelslike,  
 -name,  
 -conditions,  
 -description,  
 -stations,  
 -sunset,  
 -sunrise,  
 -datetime,  
 -sunrise,  
 -sunset,  
 -preciptype,  
 -icon,  
 -datetime,  
 -avgTemp,  
 -precipprob,  
 -severerisk)  
  
df\_base$temp <- df\_base$temp + 50  
df\_base$tempmax <- df\_base$tempmax + 50  
df\_base$tempmin <- df\_base$tempmin + 50

temp

df\_reg <- df\_base %>% select(-tempmax, -tempmin)  
  
df\_reg$windgust <- ifelse(is.na(df\_reg$windgust), 0, df\_reg$windgust)  
#df\_reg$severerisk <- ifelse(is.na(df\_reg$severerisk), 0, df\_reg$severerisk)  
  
df\_reg\_ts <- ts(df\_reg, start = c(2012, 121.25), frequency = 365.25)  
  
# Box Cox  
df\_reg\_ts\_bc <- df\_reg\_ts  
df\_reg\_ts\_bc[,"temp"] <- BoxCox(df\_reg\_ts[,"temp"], lambda = 2)  
  
# split  
train\_ts <- window(df\_reg\_ts\_bc, start = c(2012, 121.25), end = c(2021, 120))  
test\_ts <- window(df\_reg\_ts\_bc, start = c(2021, 121))  
  
train\_ts\_2 <- window(df\_reg\_ts, start = c(2012, 121.25), end = c(2021, 120))  
test\_ts\_2 <- window(df\_reg\_ts, start = c(2021, 121))

# forward stepwise  
regfit <- regsubsets(temp ~.,  
 data = train\_ts,  
 nvmax = 16,  
 method = "forward")  
summary(regfit)

## Subset selection object  
## Call: regsubsets.formula(temp ~ ., data = train\_ts, nvmax = 16, method = "forward")  
## 16 Variables (and intercept)  
## Forced in Forced out  
## dew FALSE FALSE  
## humidity FALSE FALSE  
## precip FALSE FALSE  
## precipcover FALSE FALSE  
## snow FALSE FALSE  
## snowdepth FALSE FALSE  
## windgust FALSE FALSE  
## windspeed FALSE FALSE  
## winddir FALSE FALSE  
## sealevelpressure FALSE FALSE  
## cloudcover FALSE FALSE  
## visibility FALSE FALSE  
## solarradiation FALSE FALSE  
## solarenergy FALSE FALSE  
## uvindex FALSE FALSE  
## moonphase FALSE FALSE  
## 1 subsets of each size up to 16  
## Selection Algorithm: forward  
## dew humidity precip precipcover snow snowdepth windgust windspeed  
## 1 ( 1 ) "\*" " " " " " " " " " " " " " "   
## 2 ( 1 ) "\*" "\*" " " " " " " " " " " " "   
## 3 ( 1 ) "\*" "\*" " " " " " " "\*" " " " "   
## 4 ( 1 ) "\*" "\*" " " " " " " "\*" " " " "   
## 5 ( 1 ) "\*" "\*" " " " " " " "\*" " " " "   
## 6 ( 1 ) "\*" "\*" " " " " " " "\*" " " " "   
## 7 ( 1 ) "\*" "\*" " " " " " " "\*" " " " "   
## 8 ( 1 ) "\*" "\*" " " " " " " "\*" " " " "   
## 9 ( 1 ) "\*" "\*" " " " " " " "\*" " " " "   
## 10 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " " "   
## 11 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " " "   
## 12 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" " " " "   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " " "   
## 14 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*"   
## 15 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## winddir sealevelpressure cloudcover visibility solarradiation  
## 1 ( 1 ) " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " "   
## 3 ( 1 ) " " " " " " " " " "   
## 4 ( 1 ) " " " " " " "\*" " "   
## 5 ( 1 ) " " " " "\*" "\*" " "   
## 6 ( 1 ) "\*" " " "\*" "\*" " "   
## 7 ( 1 ) "\*" "\*" "\*" "\*" " "   
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 10 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 12 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 14 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 15 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*"   
## solarenergy uvindex moonphase  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " " "   
## 4 ( 1 ) " " " " " "   
## 5 ( 1 ) " " " " " "   
## 6 ( 1 ) " " " " " "   
## 7 ( 1 ) " " " " " "   
## 8 ( 1 ) " " " " " "   
## 9 ( 1 ) " " "\*" " "   
## 10 ( 1 ) " " "\*" " "   
## 11 ( 1 ) "\*" "\*" " "   
## 12 ( 1 ) "\*" "\*" " "   
## 13 ( 1 ) "\*" "\*" " "   
## 14 ( 1 ) "\*" "\*" " "   
## 15 ( 1 ) "\*" "\*" " "   
## 16 ( 1 ) "\*" "\*" "\*"

# check error using test data  
test.mat <- model.matrix(temp ~., data = as.data.frame(test\_ts))  
  
val.error <- rep(NA, 16)  
for (i in 1:16) {  
 coefi = coef(regfit, id = i)  
 pred = test.mat[, names(coefi)]%\*%coefi  
 val.error[i] = mean((test\_ts[, "temp"] - pred)^2)  
}  
val.error

## [1] 45405.264 4427.207 4016.765 3584.241 3437.932 3323.558 3280.951  
## [8] 3380.878 3536.803 3574.952 3582.116 3575.428 3578.035 3573.702  
## [15] 3571.899 3572.910

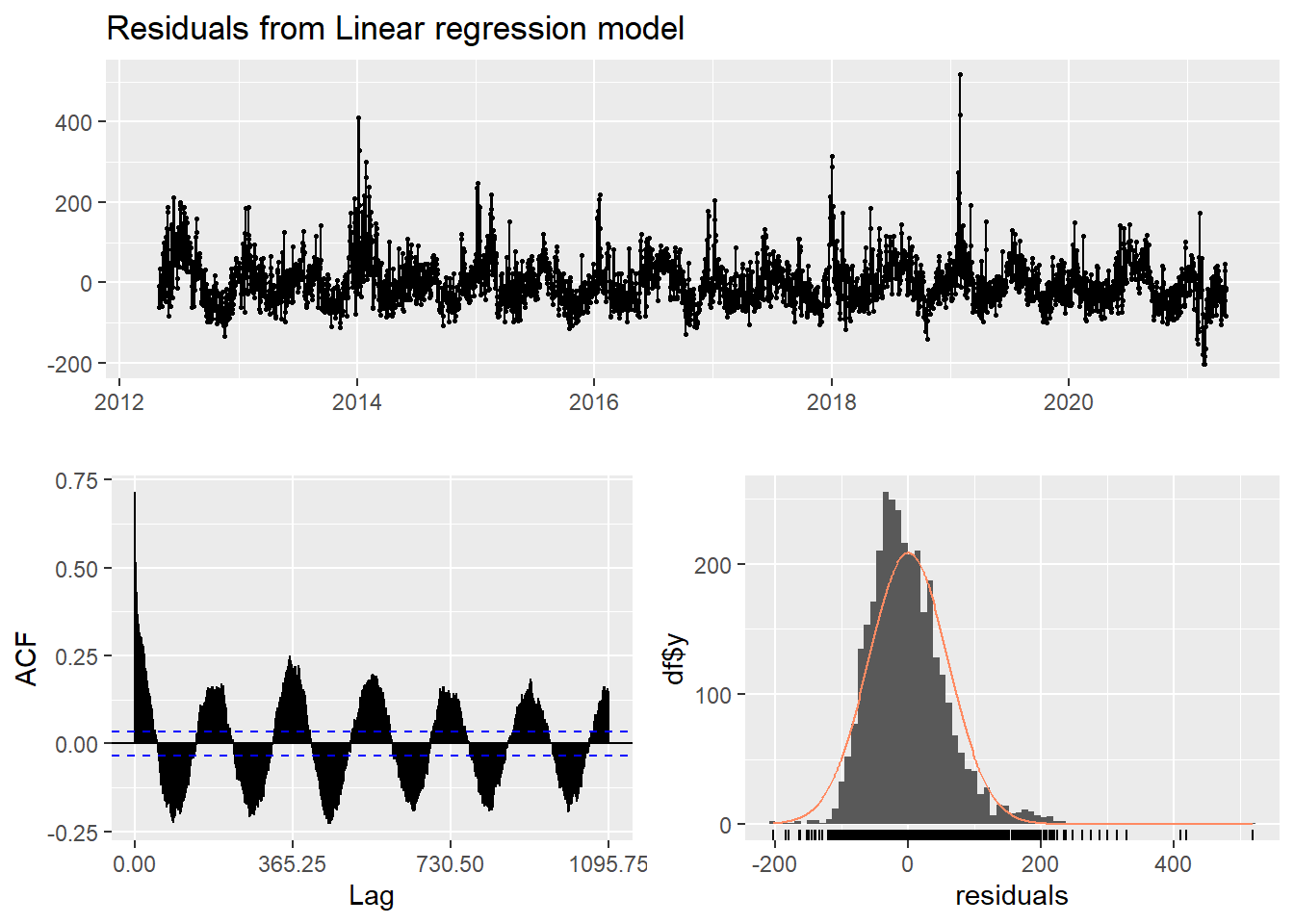
coef(regfit, which.min(val.error))

## (Intercept) dew humidity snowdepth   
## 1793.0604804 64.4995236 -16.4502534 11.0289314   
## winddir sealevelpressure cloudcover visibility   
## 0.1166410 1.0855355 -0.5622263 -10.7879655

# best model (p = 7)  
bestfit <- tslm(temp ~ dew + humidity + snowdepth +  
 winddir + sealevelpressure +   
 cloudcover + visibility,  
 data = train\_ts\_2,  
 lambda = 2)  
  
summary(bestfit)

##   
## Call:  
## tslm(formula = temp ~ dew + humidity + snowdepth + winddir +   
## sealevelpressure + cloudcover + visibility, data = train\_ts\_2,   
## lambda = 2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -203.28 -39.48 -7.77 31.67 518.96   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1793.06048 185.84566 9.648 < 2e-16 \*\*\*  
## dew 64.49952 0.13049 494.268 < 2e-16 \*\*\*  
## humidity -16.45025 0.13543 -121.468 < 2e-16 \*\*\*  
## snowdepth 11.02893 0.69563 15.855 < 2e-16 \*\*\*  
## winddir 0.11664 0.01097 10.629 < 2e-16 \*\*\*  
## sealevelpressure 1.08554 0.18183 5.970 2.63e-09 \*\*\*  
## cloudcover -0.56223 0.05492 -10.237 < 2e-16 \*\*\*  
## visibility -10.78797 0.65877 -16.376 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 59.84 on 3279 degrees of freedom  
## Multiple R-squared: 0.992, Adjusted R-squared: 0.992   
## F-statistic: 5.793e+04 on 7 and 3279 DF, p-value: < 2.2e-16

# residual  
checkresiduals(bestfit)



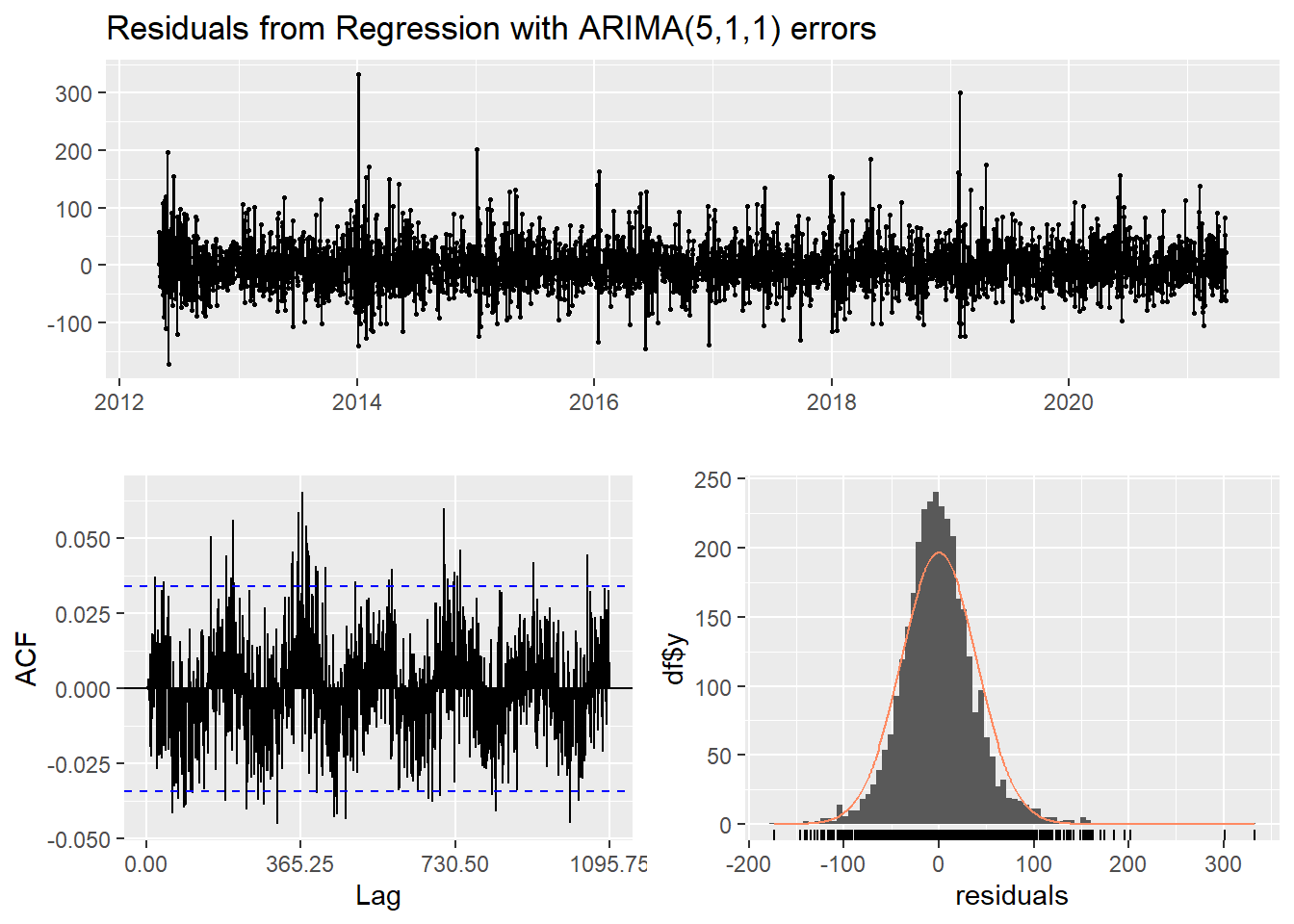
##   
## Breusch-Godfrey test for serial correlation of order up to 657  
##   
## data: Residuals from Linear regression model  
## LM test = 2080, df = 657, p-value < 2.2e-16

* Need to do regression with ARIMA errors.

xreg <- cbind(dew = train\_ts\_2[, "dew"],  
 humidity = train\_ts\_2[, "humidity"],  
 snowdepth = train\_ts\_2[, "snowdepth"],  
 winddir = train\_ts\_2[, "winddir"],  
 sealevelpressure = train\_ts\_2[, "sealevelpressure"],  
 cloudcover = train\_ts\_2[, "cloudcover"],  
 visibility = train\_ts\_2[, "visibility"])  
  
reg\_w\_arima <- auto.arima(train\_ts\_2[, "temp"],  
 lambda = 2,  
 xreg = xreg)  
  
summary(reg\_w\_arima)

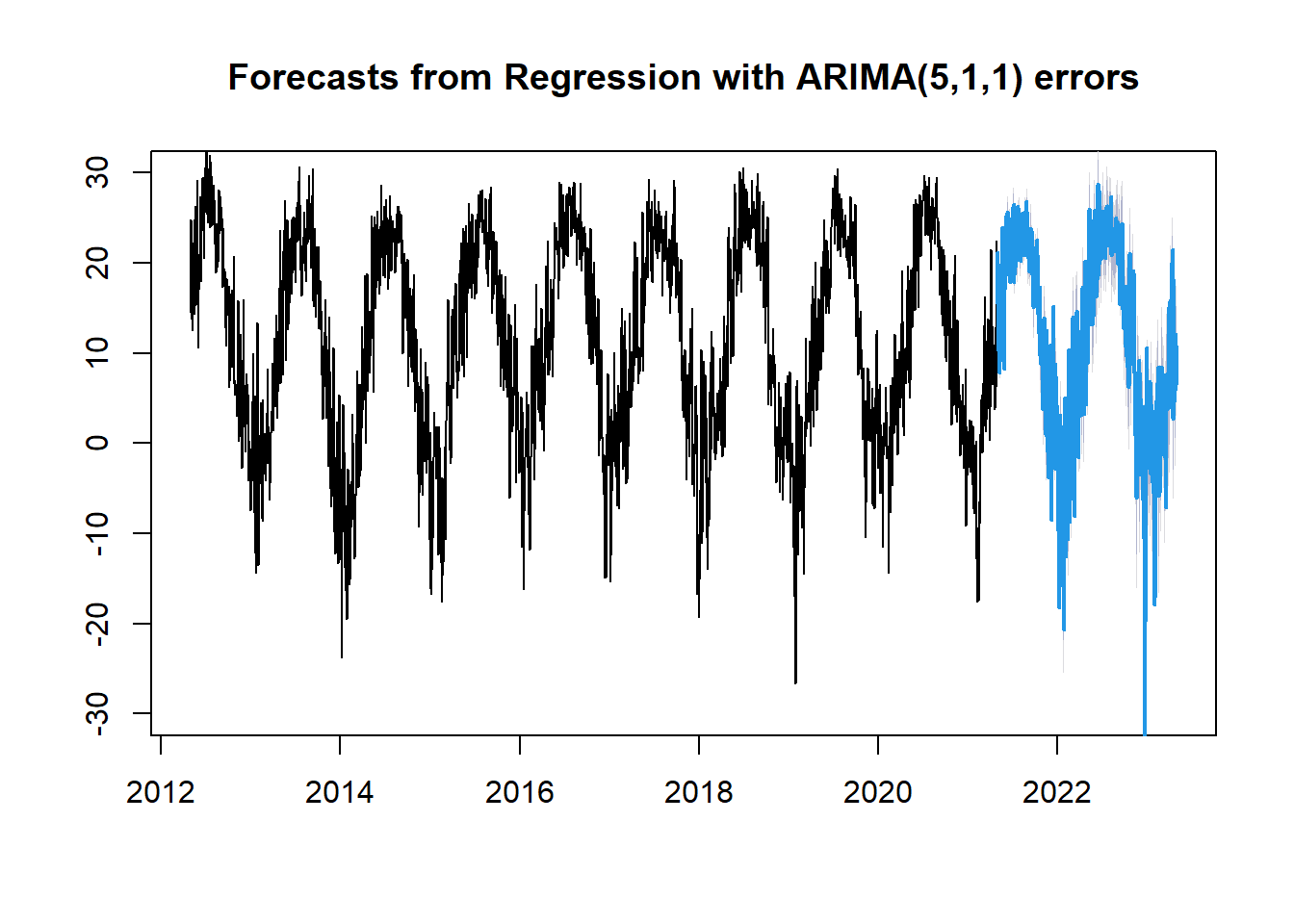
## Series: train\_ts\_2[, "temp"]   
## Regression with ARIMA(5,1,1) errors   
## Box Cox transformation: lambda= 2   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ma1 dew humidity  
## 0.6107 -0.1361 0.0103 0.0464 0.0049 -0.9066 61.9493 -16.1025  
## s.e. 0.0226 0.0211 0.0215 0.0210 0.0191 0.0141 0.2635 0.1153  
## snowdepth winddir sealevelpressure cloudcover visibility  
## 7.2025 0.0757 0.3397 -0.3648 -8.6450  
## s.e. 1.1918 0.0069 0.1685 0.0373 0.4189  
##   
## sigma^2 = 1587: log likelihood = -16764.38  
## AIC=33556.76 AICc=33556.89 BIC=33642.13  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.02559956 0.9898187 0.5211338 0.1257344 0.9942067 0.1007785  
## ACF1  
## Training set 0.07599452

checkresiduals(reg\_w\_arima)



##   
## Ljung-Box test  
##   
## data: Residuals from Regression with ARIMA(5,1,1) errors  
## Q\* = 841.18, df = 651, p-value = 6.228e-07  
##   
## Model df: 6. Total lags used: 657

newxreg <- cbind(dew = test\_ts\_2[, "dew"],  
 humidity = test\_ts\_2[, "humidity"],  
 snowdepth = test\_ts\_2[, "snowdepth"],  
 winddir = test\_ts\_2[, "winddir"],  
 sealevelpressure = test\_ts\_2[, "sealevelpressure"],  
 cloudcover = test\_ts\_2[, "cloudcover"],  
 visibility = test\_ts\_2[, "visibility"])  
fcast <- forecast(reg\_w\_arima, xreg = newxreg)  
fcast[["mean"]] <- fcast[["mean"]] - 50  
fcast[["lower"]] <- fcast[["lower"]] - 50  
fcast[["upper"]] <- fcast[["upper"]] - 50  
fcast[["x"]] <- fcast[["x"]] - 50  
fcast[["fitted"]] <- fcast[["fitted"]] - 50  
plot(fcast,  
 ylim = c(-30, 30))



test\_ts\_2\_true <- test\_ts\_2[, "temp"] - 50  
test\_ts\_2\_true

## Time Series:  
## Start = 2021.32854209446   
## End = 2023.32717316906   
## Frequency = 365.25   
## [1] 17.0 22.5 17.6 12.2 10.3 10.0 9.5 7.4 7.9 7.9 8.4 10.4  
## [13] 13.7 16.0 14.6 17.2 17.5 19.0 20.9 24.6 24.7 25.8 22.2 23.5  
## [25] 25.4 22.5 13.3 8.5 11.8 14.1 17.3 18.1 19.8 23.1 27.0 27.5  
## [37] 27.3 24.5 25.8 25.9 26.0 27.9 26.5 25.5 23.5 21.1 20.2 24.4  
## [49] 27.3 25.6 23.6 20.0 18.6 21.4 21.9 23.0 24.0 25.0 22.8 25.2  
## [61] 24.4 21.4 19.2 23.3 26.9 28.1 28.7 26.1 19.0 20.1 20.8 20.0  
## [73] 21.4 23.0 25.3 23.7 21.0 22.3 23.5 24.1 24.5 21.1 24.1 27.7  
## [85] 28.1 27.7 28.0 27.0 27.9 26.4 21.0 22.3 21.9 21.0 22.7 23.9  
## [97] 25.1 24.8 25.8 27.2 24.4 27.7 26.9 26.0 24.4 22.7 21.9 22.6  
## [109] 24.1 25.8 26.2 27.1 26.6 24.5 25.7 27.9 26.5 26.2 28.0 28.9  
## [121] 27.4 24.8 24.1 22.8 21.5 21.5 21.4 22.1 22.2 23.2 21.0 19.6  
## [133] 21.3 24.5 25.3 24.9 25.2 20.4 21.5 24.3 22.1 24.8 24.8 19.2  
## [145] 15.5 15.0 18.9 16.1 20.8 22.3 19.1 20.3 20.8 22.6 22.4 21.5  
## [157] 19.7 19.2 19.2 20.5 19.8 21.8 23.7 21.9 16.9 16.9 19.1 14.2  
## [169] 11.4 12.5 14.9 16.2 17.3 12.3 9.6 8.8 8.7 10.5 9.8 9.0  
## [181] 11.6 12.4 12.0 8.7 5.3 4.3 2.9 3.8 5.8 8.6 11.9 13.2  
## [193] 11.6 10.3 10.6 3.7 3.2 2.2 0.8 5.6 12.3 1.5 1.2 5.6  
## [205] 5.9 -1.0 1.2 7.6 4.1 -3.6 2.1 0.8 1.6 5.5 6.2 10.6  
## [217] 7.3 3.3 3.6 -1.1 -7.1 -1.5 3.4 6.6 5.6 4.5 5.3 7.2  
## [229] 15.8 8.1 1.7 3.0 1.1 2.1 1.0 -3.0 3.3 9.1 7.4 2.6  
## [241] 4.7 1.9 2.0 1.9 4.3 1.6 -4.7 -8.2 -1.2 -6.8 -11.6 -13.3  
## [253] -4.3 -4.0 -10.0 -6.0 2.9 2.8 0.5 -4.4 -5.3 -2.7 0.9 -3.9  
## [265] -10.2 -7.6 -3.8 -7.7 -6.9 -12.9 -15.3 -3.1 -6.3 -8.5 -4.5 -0.5  
## [277] 4.1 -4.7 -5.1 -7.0 -10.2 -3.2 -4.5 -1.6 2.9 -0.7 3.0 -6.9  
## [289] -8.9 -6.3 0.0 9.9 -1.7 -6.8 -8.0 2.5 5.5 2.6 -5.6 -3.0  
## [301] -3.3 -4.2 1.7 5.1 6.3 6.3 0.2 2.3 13.7 6.1 1.3 1.1  
## [313] 2.1 -2.6 -3.7 -7.5 3.7 9.8 6.1 12.5 14.0 6.8 4.9 9.3  
## [325] 15.4 12.1 11.6 6.3 5.7 0.1 -1.6 -1.6 2.7 12.3 4.7 3.0  
## [337] 3.3 4.9 8.1 8.0 11.1 6.2 3.8 4.2 10.1 14.0 14.8 17.9  
## [349] 7.3 8.5 5.0 3.6 3.2 5.9 9.0 14.8 11.3 21.4 20.9 10.7  
## [361] 6.5 3.7 8.0 14.1 16.0 12.1 10.9 8.8 9.0 9.8 10.6 13.3  
## [373] 14.6 21.6 26.8 28.0 29.1 27.7 24.8 21.1 20.2 15.4 14.6 22.0  
## [385] 23.8 15.7 14.6 13.1 14.7 19.9 22.7 14.8 18.4 24.0 27.5 27.8  
## [397] 21.0 20.5 22.3 18.1 20.6 20.7 16.8 16.9 19.3 19.8 20.1 18.9  
## [409] 24.0 31.0 32.5 29.9 26.0 19.6 22.4 27.7 31.1 29.1 25.1 27.1  
## [421] 23.8 24.3 21.4 24.0 25.6 28.7 24.7 24.9 26.5 26.6 27.7 23.8  
## [433] 25.5 24.3 22.8 24.5 26.1 26.1 23.9 22.5 21.5 22.6 23.3 26.9  
## [445] 27.9 28.3 28.4 27.9 26.1 27.0 22.8 23.6 25.0 24.2 23.3 24.4  
## [457] 25.7 26.7 26.8 27.9 25.3 26.6 29.6 27.0 24.3 21.7 23.8 23.8  
## [469] 22.0 23.7 22.5 23.1 23.2 23.2 24.0 25.6 22.5 22.9 23.2 24.0  
## [481] 25.1 23.3 22.5 22.6 26.6 25.6 23.8 24.0 26.2 26.4 25.1 22.2  
## [493] 21.2 23.0 23.1 23.6 23.8 24.3 19.8 15.6 20.3 20.3 22.6 24.0  
## [505] 25.3 26.0 22.7 24.0 26.0 15.7 14.7 16.9 17.0 14.3 12.5 12.2  
## [517] 12.8 14.2 15.8 15.9 14.6 15.5 17.8 17.7 10.2 9.2 13.7 15.6  
## [529] 15.9 15.5 9.1 9.0 7.8 9.6 5.4 5.8 5.6 8.3 15.5 20.3  
## [541] 21.9 21.9 15.9 9.3 8.6 11.0 11.5 12.8 13.6 14.0 15.5 18.0  
## [553] 19.6 14.8 11.8 9.1 9.6 15.1 19.3 7.1 1.3 0.4 1.4 1.9  
## [565] 1.2 -0.3 -3.7 -5.4 -4.5 2.5 3.9 7.1 7.4 5.8 6.9 6.0  
## [577] 3.8 9.4 -3.1 -2.7 7.0 -1.0 -1.5 2.5 4.9 4.5 3.1 2.7  
## [589] 3.4 3.5 3.8 2.6 4.5 1.6 -1.5 -4.3 -7.8 -7.3 -2.4 -4.5  
## [601] -6.2 -20.1 -14.4 -12.9 -7.8 -8.2 2.6 10.8 6.1 0.5 5.1 3.4  
## [613] 6.1 2.8 0.6 -1.5 -1.0 -0.9 1.2 5.2 7.1 3.4 -0.7 -2.7  
## [625] 1.3 6.8 4.9 3.2 3.6 0.5 -1.1 -0.2 -2.8 0.1 0.6 -3.8  
## [637] -4.2 -4.0 -3.9 -10.9 -13.8 -8.0 -4.0 -13.5 -2.1 2.9 1.7 4.6  
## [649] 1.7 3.8 1.3 1.4 4.9 6.2 7.9 6.3 0.5 -4.8 2.2 6.7  
## [661] 3.6 0.9 1.4 1.2 -4.3 0.0 3.1 8.0 5.7 8.1 2.5 2.3  
## [673] 3.6 4.6 7.0 4.0 4.6 3.3 1.4 2.2 1.2 -1.1 -1.9 3.3  
## [685] 7.2 1.6 -6.4 -1.5 4.9 6.8 7.4 5.6 4.7 3.6 4.5 4.3  
## [697] 4.7 3.0 4.8 15.3 4.8 6.4 11.1 11.1 15.7 6.0 5.9 10.0  
## [709] 13.4 15.9 19.5 22.1 22.0 21.5 22.7 9.4 2.6 7.7 11.2 18.6  
## [721] 11.2 5.9 4.3 7.1 5.8 5.7 10.8 12.0 10.1 6.6 6.7

accuracy(fcast, test\_ts\_2\_true)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.02559956 0.9898187 0.5211338 NaN Inf 0.1007785 0.07599452  
## Test set 0.68063335 1.3072627 0.9426077 NaN Inf 0.1822845 0.66420542  
## Theil's U  
## Training set NA  
## Test set 0